Optimizer

Gradient Descent

Find the steepest direction to increase Loss and take negative direction of it to reduce loss
It becomes `Stochastic Gradient Descent`, if use single-point batch
And becomes `Mini-batch GD`, if use a few number of data less than population

$$g_t = \nabla_\theta f_t(\theta_{t-1})$$
$$\theta_t = \theta_{t-1} - \gamma \cdot g_t$$

RMSProp

Uniformly applying learning rate for every single parameter would not make sense.
What about letting dynamic, adaptive learning rate per parameter so that it can handle oscillation or vanishment.

$$g_t = \nabla_\theta f_t(\theta_{t-1})$$
$$v_t = \beta v_{t-1} + (1-\beta) g_t^2$$
$$\theta_{t,j} = \theta_{t-1,j} - \frac{\gamma}{\sqrt{v_{t,j} + \epsilon}} g_{t,j}$$

Holding other components, large value of current gradient gets partially canceled out by denominator, while boosted when gradient's small. In addition, the extent (scale) of an update for a parameter does not solely depend upon current gradient as it's using moving average.

Adam = RMSProp + Classical Momentum

In RMSProp, we saw it's using g_t as-is, which means we use current gradient but scale it using adaptive learning rate ($1/\sqrt{v_t + \epsilon}$). In Adam, we explicitly use the trend of gradient.

$$g_t = \nabla_\theta f_t(\theta_{t-1})$$
$$\text{If,} \lambda \neq 0, g_t = g_t + \lambda \theta_{t-1}$$
$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$
$$\widehat{m_t} = m_t / (1-\beta_1^t)$$
$$\widehat{v_t} = v_t / (1-\beta_2^t)$$
$$\theta_t = \theta_{t-1} - \frac{\gamma}{\sqrt{\widehat{v_t} + \epsilon}} \widehat{m_t}$$

AdamW

In Adam, regularization was part of g_t, having different impact for each parameter due to adaptive characteristics of learning rate calculation. This also means weight decaying (regularization) does not exactly work how intended but gets affected by statistics of gradients.
Thus, AdamW decoupled it from gradient. So do not include the term when calculating gradient but do only when updating parameters

$$g_t = \nabla_\theta f_t(\theta_{t-1})$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

$$\widehat{m_t} = m_t / (1-\beta_1^t)$$

$$\widehat{v_t} = v_t / (1-\beta_2^t)$$

$$\theta_t = \theta_{t-1} - \frac{\gamma}{\sqrt{\widehat{v_t}+\epsilon}} \widehat{m_t} - \gamma\lambda\theta_{t-1}$$

Practically AdamW tends to perform better but as always tests are needed for individual use case.

Personal Experiment